

Application No.: 10/827,527**Docket No.: 200400476-2 (1509-499)****AMENDMENTS TO THE CLAIMS:**

The following listing replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. *(Currently amended)* A method of generating code for scheduling the execution of binary code translated from a source format to a target format, said method comprising the steps of:

(a) identifying a set of target instructions semantically equivalent to a given source instruction;

(b) ~~analysing the set of target instructions to identify~~ identifying data dependencies in said target instructions by analyzing the set of target instructions; and

(c) assigning an identifier to one or more of said target instructions for use by a code analyser in scheduling the processing of said set of target instructions in accordance with the identified data dependencies.

2. *(Currently amended)* A method according to claim 1 in which the set of target instructions is identified in a translation template associated with a given source instruction, said template being a component of a translator program for translating instructions in the source format into instructions in the target format.

3. *(Original)* A method according to claim 2 in which the analysis of the target instructions is carried out prior to the compilation of the translation templates into said translator program.

Application No.: 10/827,527**Docket No.: 200400476-2 (1509-499)**

4. (Currently amended) A method according to claim 2 in which the identifiers are assigned to said target instructions prior to ~~the compilation of~~ said translator program being compiled.

5. (Currently amended) A method according to claim 1 in which said code analyser ~~uses the identifiers for optimising~~ optimizes the translated code for processing in a parallel processing environment by using the identifiers.

6. (Currently amended) A method according to claim 1 in which data dependencies are represented by a directed acyclic graph, and the identifier ~~is arranged to identify~~ identifying step identifies said dependency signalling an appropriate edge in the set of target instructions to said code analyser.

7. (Currently amended) A method according to claim 2 in which each translation template is associated with a corresponding analysis routine ~~for generating~~ that generates said code for scheduling the execution of said translated code.

8. (Currently amended) Apparatus for generating code for scheduling the execution of binary code translated from a source format to a target format, said apparatus comprising:

(a) a set of target instructions semantically equivalent to a given source instruction;

(b) an instruction analyser for analysing the set of target instructions to identify data dependencies in said target instructions; and

Application No.: 10/827,527**Docket No.: 200400476-2 (1509-499)**

(c) a dependency identifier for assigning an identifier to one or more of said target instructions for use by a code analyser in scheduling the processing of said set of target instructions in accordance with the identified data dependencies.

9. *(Currently amended)* Apparatus according to claim 8 ~~in which the set of target instructions is identified in~~ further including a translation template for identifying the set of target instructions, the translation template being associated with a given source instruction, said template being a component of a translator program for translating instructions in the source format into instructions in the target format.

10. *(Currently amended)* Apparatus according to claim 9 ~~in which the analysis of wherein the instruction analyzer is arranged for arranging the target instructions is carried out prior to~~ [[the]] compilation of the translation templates into said translator program.

11. *(Currently amended)* Apparatus according to claim 9 ~~in which the identifiers are assigned~~ wherein the dependency identifier is arranged to assign the identifier to said target instructions prior to [[the]] compilation of said translator program.

12. *(Currently amended)* Apparatus according to claim 8 in which said code analyzer [[uses]] is arranged to use the identifiers for optimising the translated code for processing in a parallel processing environment.

13. *(Currently amended)* Apparatus according to claim 8 ~~in which~~ further including a directed acyclic graph for representing data dependencies, are represented

Application No.: 10/827,527**Docket No.: 200400476-2 (1509-499)**

~~by a directed acyclic graph~~ and the identifier is arranged to identify said dependency signalling an appropriate edge in the set of target instructions to said code analyser.

14. (*Original*) Apparatus according to claim 9 in which each translation template is associated with a corresponding analysis routine for generating said code for scheduling the execution of said translated code.

15. (*Currently amended*) A ~~computer program arranged~~ computer readable medium or storage device storing coded indicia for causing a data processor arrangement to perform the method of claim 1.

16. (*Currently amended*) A binary code translator for translating binary code from a source format to a target format for execution on a target processor, the translator comprising a computer-readable medium or storing device storing coded indicia adapted to be read by a data processor arrangement, the coded indicia including:

(a) a set of translation templates, each template arranged for providing a set of target format instructions which together are semantically equivalent to an associated source format instruction;

(b) a set of data transformation routines arranged to transform data from a source format instruction into the appropriate parts of each target format instruction provided by the corresponding translation template; and

Application No.: 10/827,527**Docket No.: 200400476-2 (1509-499)**

(c) a set of analysis routines arranged to identify data dependencies in a template for ~~generating~~ causing generation of data for use by a code scheduler in scheduling the execution of translated code on said target processor.

17. (*Currently amended*) A binary code translator according to claim 16 arranged to operate dynamically at the run time of an application program being emulated.

18. (*New*) The method of claim 1 wherein the code analyzer schedules the processing of said set of target instructions in accordance with the identified data dependencies.

19. (*New*) The apparatus of claim 8 in combination with the code analyzer arranged to be responsive to the identifier assigned to one or more of the target instructions, the code analyzer being arranged for scheduling the processing of said set of target instructions in accordance with the identified data dependencies.

20. (*New*) Apparatus for translating machine instructions in source code into equivalent target instructions of a code of a target platform, wherein the source code differs from the code of the target platform, said apparatus comprising:

a source of binary translation templates for mapping instructions in the source code into a set of instructions in the code of the target platform;

a fill and analysis routine generator arranged to be responsive to the templates for generating fill and analysis routines for identifying fillable positions in a template by

Application No.: 10/827,527

Docket No.: 200400476-2 (1509-499)

parsing the template and for generating code to extract and deposit fields from the machine instructions in source code into a precompiled template; and

a dynamic binary translator arranged to be responsive to the machine instructions.